A configuration assistant for versatile vision-based inspection systems

Olivier Hüsser
Institute of Microtechnology
University of Neuchâtel
Rue A.-L. Breguet 2
2000 Neuchâtel
olivier.huesser@imt.unine.ch

Heinz Hügli
Institute of Microtechnology
University of Neuchâtel
Rue A.-L. Breguet 2
2000 Neuchâtel
heinz.hugli@imt.unine.ch

ABSTRACT

Nowadays, vision-based inspection systems are present in many stages of the industrial manufacturing process. Their versatility, which permits to accommodate a broad range of inspection requirements, is however limited by the time consuming system setup performed at each production change. This work aims at providing a configuration assistant that helps to speed up this system setup, considering the peculiarities of industrial vision systems. The pursued principle, which is to maximize the discriminating power of the features involved in the inspection decision, leads to an optimization problem based on a high dimensional objective function. Several objective functions based on various metrics are proposed, their optimization being performed with the help of various search heuristics such as genetic methods and simulated annealing methods. The experimental results obtained with an industrial inspection system are presented. They show the effectiveness of the presented approach, and validate the configuration assistant as well.

Keywords: Visual inspection, vision systems, automatic setup, system configuration, heuristic search, decision, objective function

1. INTRODUCTION

In the automatic inspection field, advanced visual inspection systems perform a large and ever growing number of controls. These systems, however, require some tuning in order to perform efficiently on a specific task. Given the increasing complexity of the system, manual tuning becomes a major problem of an operator using such a system. Means for automatic tuning are required.

Automatic optimization tasks in the vision field are numerous. Some example tasks recently investigated are: contour tracking with *snakes*[1]; pattern recognition in association with *markov random field* representations[2]; image restoration[3]; geometric transformation estimation by *iterative closest point* methods[4]. Among the few papers related to visual inspection systems and their optimization, the paper by Daniel and West[5] describes a method for modeling shape variations of forged devices from samples. Using eigenvectors and an optimization algorithm based on an energy function he describes a way to unveil the mapping between process control parameters and shape deformation of the forged device allowing to optimize the production process. In the present paper, the focus is set on the optimization of a visual inspection system in order for it to minimize its error rate.

A visual inspection system consists mainly of two parts: an image analysis and a decision part. The image of the device to be inspected, Y, is first analyzed with help of image analysis tools that produce a set of measurements **M**. This measurement vector is then classified by the decision module giving the final decision ω. Both steps may involve several parameters **P** that influence the output (decision) of the system.

In order to tune the system, human operators execute the following supervised learning procedure. First, a certain number of good and bad representatives is drafted among the devices to be inspected. Then the visual inspection is performed and, depending on the system's decision, some parameters might be modified. Inspections and modifications are performed until experimental decisions match the a priori good/bad partition.

This setup process is not easy and can be time consuming. Furthermore it is approximate, in the sense that no hint on the confidence of decisions is available. In order to remedy these shortcomings, we developed a configuration assistant that is presented in this paper. The configuration assistant is designed to perform the parametric tuning automatically. The pursued idea is not only to enhance the decision module, as it is often the case, but also to maximize the discriminating power of the measurements.

The performance of the visual inspection must be formulated in a way that permits a comparison between the expected and experimental results. Somehow, this comparison must be quantifiable and the resulting quantity be pertinent. We call this quantity the **objective function F**. This scalar function over the parameters of the system should trustfully represent the realization of the expected results, must be reliable and informative altogether. Thus, in order to find the best configuration for the system the idea is to use the objective function to measure its performance and to use search heuristics for exploring the parametric space spanned by **P**. Practically, a "good enough" solution is sufficient.

This paper focuses on the design of a configuration assistant for an inspection system whose decision module consists of a so-called *and-classifier*. This type of decision is widespread in the industrial inspection field where checked devices have to satisfy multiple mandatory constraints.

The paper is organized as follows. Next section describes the inspection system. We present in section 3 some ways to define the objective function appropriate for an *and-classifier* and present the methods used to perform the supervised learning. Section 4 presents the particular case of the configuration assistant applied to IC marking inspection. Results obtained with real data are presented in section 5.

2. INSPECTION SYSTEM

Consider the inspection system of figure 1 which visually analyzes the image Y and provides a decision for an object. The analysis of the image may involve several parameters **P** that influence the resulting measurements **M**. The decision system using the measurements takes a decision ω about the inspected device. The decision rule may also have some degrees of freedom **T** that influence the outcome.

Industrial visual inspection systems determine whether a produced device satisfies some requirements. Most of the time, all of these requirements have to be satisfied in order for a device to be labeled as good. These requirements are said to be mandatory: a device must satisfy the first requirement *and* the second *and* all others to be accepted. Hence the name of *and-classifier* used here to designate the decision stage.

Given the definition of figure 1 the requirements are formulated with respect to some measurements **M** and usually take the form of thresholds T_i. In addition, a convenient inspection system, has normalized measurements with high values for good devices. The decision rule of such a classifier is written as:

$$\omega = \begin{cases} \text{accepted} & \text{if } \mathrm{M_i} > T_i \ \forall\, \mathrm{i}, \mathrm{i} = 1,2,...,\mathrm{n} \\ \text{rejected} & \text{otherwise} \end{cases} \tag{1}$$

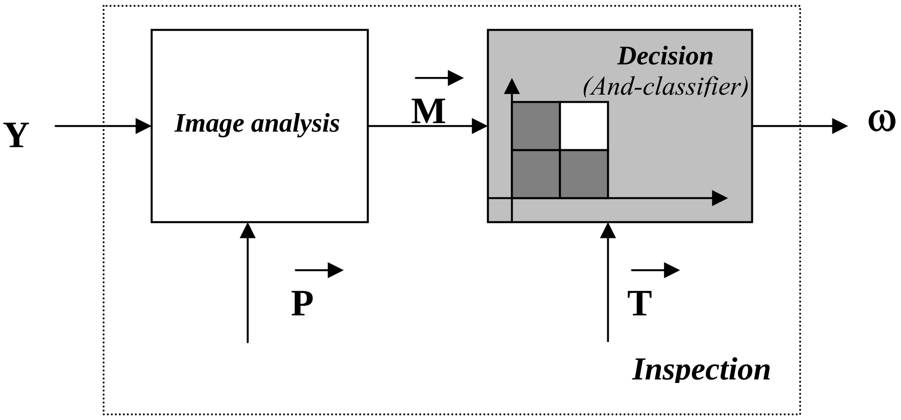

Figure 1: Structure of an inspection process

Given a set of devices represented by their digital image Y={Y_1, Y_2, …, Y_n }, and given an a priori classification of this set of devices by a human operator ω^*(Y)={ ω^*(Y_1), ω^*(Y_2), …, ω^*(Y_n)} the **learning set** is defined as Ω_L={Y, ω^*(Y)}. The learning set is the union of two subsets: the good and the bad device's subsets. They are denoted here by Ω_x (good) and Ω_o (bad).

Given the learning set, the goal is to find parameters **P** and **T** such that expected inspection performance on future objects is optimized. Next section presents objective functions as a measure for the expected performance as well as search methods for finding the optimum.

3. OBJECTIVE FUNCTIONS AND SEARCH METHODS

This section presents objective functions F used to measure the quality of the parameter's configuration in a system whose decision stage is an *and-classifier*. It then introduces the search methods used to find the maximum of F, i.e. to find the best parametric configuration according to F.

3.1. Objective Functions

To optimize the expected system's performance in a general context, the available learning set will be used. The goal is then to optimize the system's performance over this learning set. A first and obvious way to evaluate this performance is the **recognition rate**, R, defined as the fraction of correct decisions. It is normalized and it takes its maximum value of 1 if and only if all devices of the learning set are correctly classified. Although R represents the aptitude of the system on the learning set, it is not a good function for practical use. First because it is quantified on a small number of levels (=Card{Ω_L}) and is thus a rather rough function of the configuration. Furthermore it gives no hint on the confidence in the decision taken by the system. The main idea on the way to find a better performance measurement is to consider rather a function of the learning set's measurements F({M}$_{\Omega L}$) than a function of the learning set's partition F(ω).

Two hypotheses about the system are made. First, it is believed that the learning set has been selected with care. Secondly, the performed image analysis is supposed to be adequate to provide discriminant measurements. If these two conditions are satisfied, then it is possible using the degrees of freedom of the system to find a configuration that performs faultless with the learning set (R=1). The optimal margin classifier idea used by Vapnik [6] has been employed here in order to minimize the risk of a sample's misclassification over the learning set. It is also expected that this approach ensures a stable system behaving faultless with new devices (generalization) as long as they have representatives within the learning set.

3.1.1 Objective function for 1-dimensional measurements

Figure 2 displays the measurements M_y of a learning set of devices for a 1-dimensional pseudo *and-classifier*. The "x" symbols are the measurements of the Ω_x subset. Conversely, the "o" symbols are the measurements of the Ω_o subset. The vertical line denotes a fixed threshold T that has been preset.

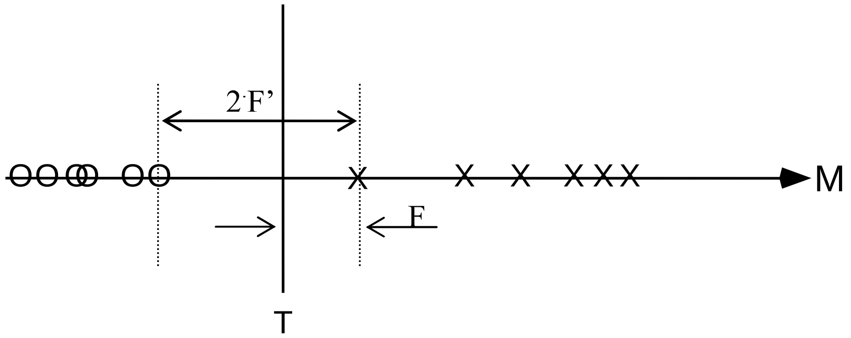

Figure 2: Objective function for 1-dimensional measurement space

The objective function F for the optimal margin is defined here as the minimum distance between a measurement M_y and the threshold T conditional to an absolutely correct classification:

Given the measurements margin $\delta(M_Y,T)=\begin{cases}\max(0,M_Y-T) & \text{for } Y\in\Omega_x\\ \max(0,T-M_Y) & \text{for } Y\in\Omega_o\end{cases}$. (2)

The objective function F takes the form $F = \min_{Y\in\Omega_L} \delta(M_Y, T)$ (3)

If the threshold is variable it can be optimized. The best position is at mid-distance of the closest pair of bad/good measurements. According to it, the objective function takes the form

$$F' = \frac{1}{2}\cdot\max(0;\min_{Y\in\Omega_x} M_Y - \max_{Y\in\Omega_o} M_Y)\,. \tag{4}$$

Note that the value of both F and F' is 0 when good and bad devices are not separable.

3.1.2 Objective function for n-dimensional measurements

The objective function for multidimensional *and-classifiers* relies on the same concepts as for the 1-dimensional case. Some definitions must be given in order to define its variants.

Given the **decision space** D spanned by the set of measurements, we define the **acceptance space** A of the classifier as the subspace of D where the inspection system considers the measurements as belonging to an acceptable device. In the complementary **rejection space** the system rejects the devices: $\overline{A} = D - A$. The partition in $\overline{A}$ and A of an *and-classifier* is defined by the n thresholds.

The objective function should reflect how far the measurements are from a wrong classification, i.e. from the wrong, or opposite, subspace. The minimum distance is still the main concern and so, the following **objective function** F is defined based on above definition: *F is equal to the minimum distance, over the learning set, of a measurement to its opposite subspace.*

This can be either the distance of a good device to $\overline{A}$, or the one of a bad device to A:

$$F = \min\left[\min_{Y\in\Omega_o}\left\|M_Y;A\right\|;\min_{Y\in\Omega_x}\left\|M_Y;\overline{A}\right\|\right], \tag{5}$$

where "|| ||" denotes the minimum distance between a point and a subspace. Note that "|| ||"=0 when the point belongs to the subspace. The notion of distance "|| ||" is related to a metric. The distance functions take the following generic form for a L_n-metric: $\|\ \|_n = \left(\sum \Delta^n\right)^{\frac{1}{n}}$. In one-dimensional case all metrics are equivalent, and so the definition of F is unique. This is not the case in n-dimensional spaces where objective functions will take distinct forms depending on the chosen metric. Figure 3 shows in 2-dimensional space the equipotential lines of F defined with 3 different metrics: L_1, L_2, L_∞. It is visible that they only differ when multiple measurements compose the distance.

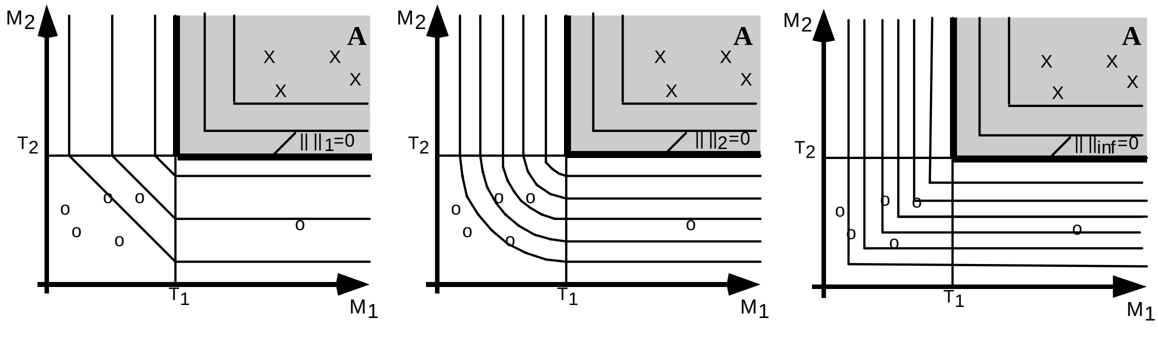

Figure 3: Acceptance subspace & contour plot of objective functions based on L_1, L_2, L_∞

Given the good devices margin, defined as the minimum distance of any good device to $\overline{A}$:

$$\delta_x = \min_{Y\in\Omega_x}\Big[\min_i\big[\max(0;M_{Yi} - T_i)\big]\Big] \tag{6}$$

and given the margin components δ_{Yi} for a bad device Y and component i:

$$\delta_{Yi} = \max(0; T_i - M_{Yi}) \ , \ Y \in \Omega_o \tag{7}$$

(5) is rewritten with different metrics and fixed thresholds T_i:

$$F_1 = \min\left[\min_{Y\in\Omega_o}\left(\sum_i \delta_{Yi}\right);\delta_x\right] \tag{8}$$

$$F_2 = \min\left[\min_{Y\in\Omega_o}\left(\sqrt{\sum_i {\delta_{Yi}}^2}\right);\delta_x\right] \tag{9}$$

$$F_\infty = \min\left[\min_{Y\in\Omega_o}\left[\max_i(\delta_{Yi})\right];\delta_x\right] \tag{10}$$

If the thresholds are variable they can be optimized according to the measurements of Ω_L. The resulting acceptance subspace A must include all Ω_x measurements and exclude all Ω_o measurements. To formally define this constraint and have a convenient way to calculate the objective function, we define here a minimum acceptance space A^*. Given the smallest measurement components of the good subset

$$M_{xi}^{\min} = \min_{Y\in\Omega_x}(M_{Yi}) \tag{11}$$

A^* is the subspace whose points **P** satisfy $P_i \geq M_{xi}^{\min}\ \forall \mathrm{i}$.

So, the F' objective function is defined similar to the 1-dimensional case as half of the minimum distance of all Ω_o measurement to A^*.

$$F' = \frac{1}{2}\cdot\min_{Y\in\Omega o}\left\|M_Y; A^*\right\| \tag{12}$$

Figure 4 displays acceptance spaces for a bi-dimensional classifier. Left hand displays the acceptance space A used with fixed thresholds and right hand displays the minimum acceptance space A^* used with variable thresholds.

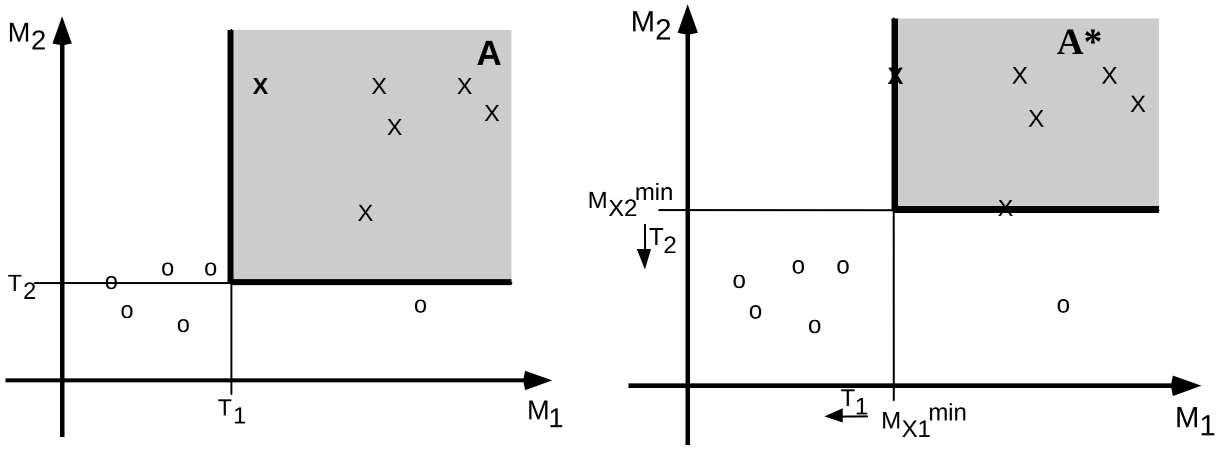

Figure 4: Acceptance subspaces A and A with fixed (left) or tunable thresholds (right)*

given the double-margin components :

$$\delta'_{Yi}{}' = \max\left(0; M_{xi}^{\min} - M_{Yi}\right), \tag{13}$$

the (8), (9), (10) equivalent functions with configurable thresholds are :

$$F_1' = 0.5 \cdot \min_{Y\in\Omega_o}\left(\sum_i \delta'_{Yi}{}'\right) \tag{14}$$

$$F_2' = 0.5 \cdot \min_{Y\in\Omega_o}\left[\sqrt{\sum_i \delta_{Yi}^{\prime 2}}\right] \tag{15}$$

$$F_\infty' = 0.5 \cdot \min_{Y\in\Omega_o}\left[\max_i(\delta'_{Yi}{}')\right] \tag{16}$$

3.2. Search Methods

Objective functions described before are defined over the measurements of the learning set which are depending on the parameters of the configuration. This dependency can take any form and is not restricted to be analytical. The number of parameters is not specified either. So practically objective functions of unknown and different nature will be found. The difficulty is to maximize over the set of possible configurations an objective function F (**P**,**T**) of unknown nature. To do this we will use methods that are recognized as robust and efficient in such situations: *genetic algorithms* and *simulated annealing*. A third method called *enhanced random* is presented and also considered.

One concern with the employed methods, is to avoid the replication of the problem to be solved. What cannot be tolerated, is a search method that would require some choices in order to function properly. It would just shift the parametric configuration problem to another level. Search methods must therefore be designed with versatility in mind in order for them to cope reasonably well with various tasks.

3.2.1 Genetic algorithm (GA)

For a general description of genetic algorithms we refer to existing documents [7, 8] and will restrict ourselves to the description of the particular implementation of the *simple genetic algorithm* used in this work. Its initial population is selected randomly over {**P**}. The evaluation, or fitness function, is the objective function previously discussed. The *roulette wheel* selection scheme is applied. A *non elitist* replacement scheme associated with a high mutation rate (P_{mut}=0.1) and an explorative *crossover* method prevent the GA's stagnation. We applied two crossover methods. First a standard multiple point crossover with a probability of 10% and more frequently (90%) a linear crossover scheme. The linear method creates two offspring located on the extended segment that connects both parents in {**P**} space. This process is illustrated in figure 5. The probability density is constant all along the segment.

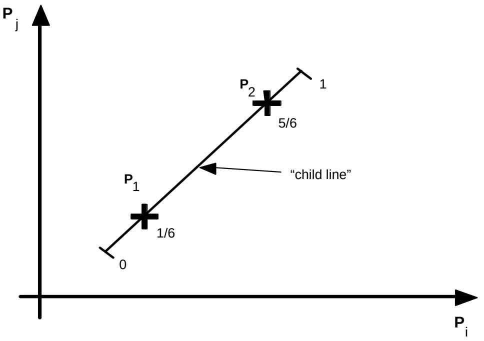

Figure 5: linear crossover process of the GA

Population size is constant and small: 20 individuals. This choice allows a large number of generations to take place until the time limit stopping criterion is reached.

3.2.2 Simulated annealing method (SA)

The name of this method comes from its relationship with physical systems and their equilibrium states. The configuration is considered here as a set of possible states for the inspection system. Associated with each state is a scalar value called *energy*. Here, the energy of a state **P** is ε(**P**)=1-F(**P**). Transition from one state to another is given a probability that depends on their relative energy values and a parameter called *temperature*. The algorithm performs at each step such a transition. This process is called **stochastic relaxation** [3]. The chosen algorithm works with a binary temperature schedule, alternating T=0 and T=∞. Figure 6 shows the flow diagram of the SA method: Starting from a randomly selected configuration (T=∞), **gradient descent** (T=0) is performed until a local minimum is reached; then if the stopping criterion is not satisfied, the process starts over.

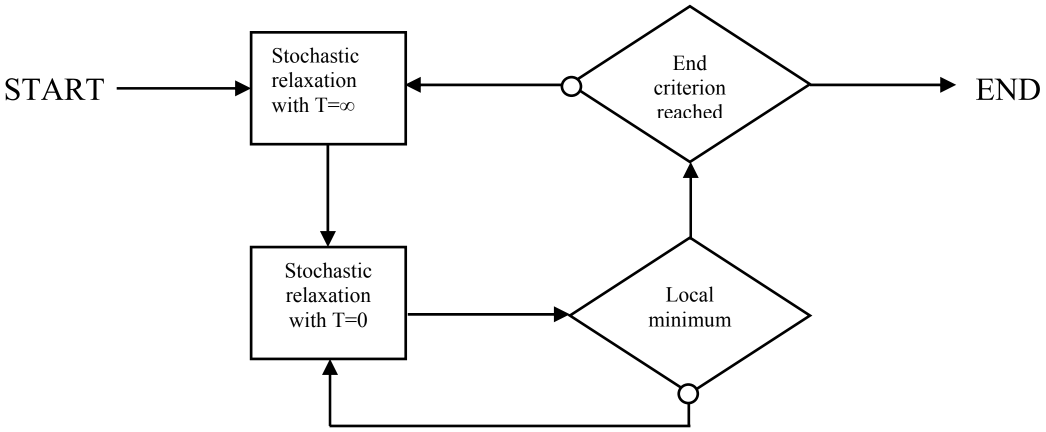
START
Stochastic relaxation with T=∞
End criterion reached
END
Stochastic relaxation with T=0
Local minimum

Figure 6:Temperature schedule of SA

3.2.3 Enhanced random method (ER)

This method combineds a random selection scheme with a probability density estimation.

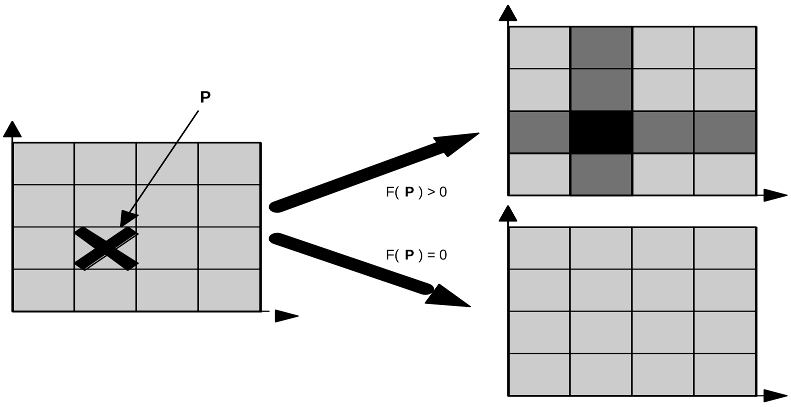

*Figure 7: modification of ρ(**P**)*

The principle of this algorithm is to use the past results to increase the probability of finding a good solution in the future. At the beginning, as no result is available, the first parametric vector **P** is chosen randomly over the whole parametric space of the permitted values specified by the user for the hidden configuration {**P**}. This random process is based on the assumption that all points $\mathbf{P}_i$ in {**P**} have the same probability to be a good solution and therefore to be chosen. In other words, the probability density ρ (**P**) is uniform over {**P**}. Then, to enhance the efficiency, the probability density is modified according to the results already obtained. This process uses the *Parzen windows*[9] technique to approximate an unknown density function of a data set over a variable. It uses a simplified $\rho(\vec{P}) = \prod_i \rho_i(p_i)$ that assumes the independence of the parameters. This gives a star-shaped modification of ρ (**P**) with a maximum at center (see figure 7). Of course, a maximum limit has to be set for ρ (**P**) in order to prevent a possible automatic selection of good tested settings.

4. AN IC MARKING INSPECTION SYSTEM

The experimental part is performed on an industrial inspection system in the context of visual inspection of integrated circuits markings. The inspection process checks the correctness of markings found on top of molded integrated circuit packages. These markings are either painted or laser burnt on the chip. An example of marking is shown in figure 8. The production environment is characterized by a high device throughput and a limited marking precision. In order to deal with these constraints, the visual inspection system possesses a number of efficient and versatile measurement and analysis tools that must be configured for optimal operation.

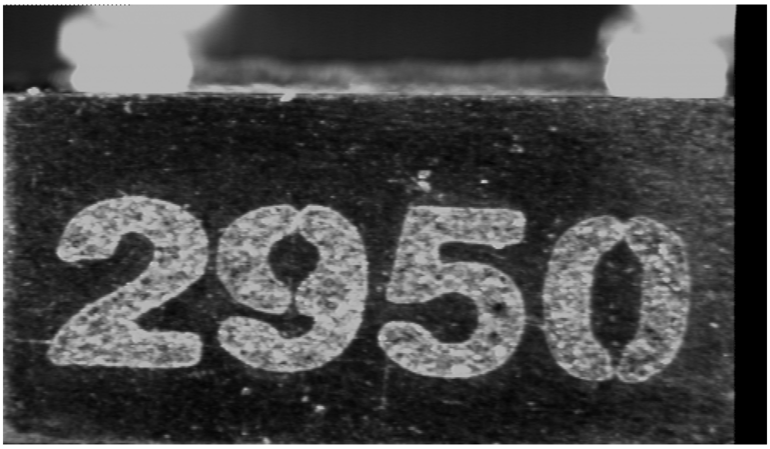

Figure 8: SOT device inspected in a production line

Decision to accept or reject a device for an eventual marking defect is based on two measurements : **contamination factor** M_1 and **missing factor** M_2. These measurements are calculated relatively to a reference (golden unit). Given Y and R the test and reference markings, we define the intersection pattern I=R∩Y(see figure 9). With the associated surfaces S_Y , S_R, and S_I the measurements are written:

$$M_1 = \frac{S_I}{S_X} \ , \ M_2 = \frac{S_I}{S_R} . \qquad (17), (18)$$

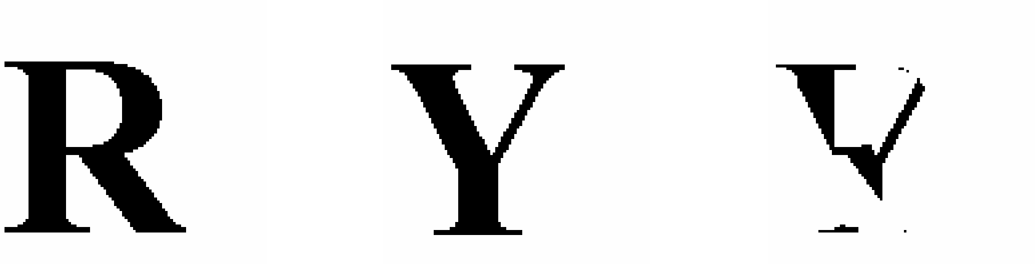

Figure 9: reference, test, and intersection patterns

Both measurements are normalized. The *contamination factor* represents the fraction of the inspected marking's surface that matches the reference and *missing factor* the fraction of the reference's surface that is covered by the inspected device. Of course, reference and test are superposed during a parametrized preprocessing step.

These measurements are simple. Their computation is fast and allows a high inspection throughput. They are versatile and apply to a broad spectrum of markings. Their disadvantage is that it is not obvious to set limit value for the acceptance of a marking. This led us to consider measurements thresholds as belonging to the configuration of the system, and accordingly to set their values with the configuration assistant. A device's marking is rejected unless ($M_1 > T_1$) and ($M_2 > T_2$).

The configuration associated with these measurements has 6 parameters. With the two configurable decision thresholds, it makes an **8-dimensional** parametric space **P**={$P_{1\text{-}6}$;$T_{1\text{-}2}$}.

Most of these parameters are concerned with the image processing of the inspected device. The effect of these parameters on the measurements is too complex to be analyzed. For example, P_1 and P_2 are offset values for the grayscale thresholds used when binarizing the reference and inspected images. The table below shows that the variation of P_1 and P_2 affects M_1 and M_2 in a way that is difficult to describe.

	S_R	S_X	S_I	M_1	M_2
P_1	↘	**-**	↘	↘	**?**
P_2	**-**	↘	↘	**?**	↘

Table 1 : Influence of system parameters on measurements

In essence it shows how increasing values of P_1 and P_2 affect the parameters. The arrow ↘ represents a decreasing value and **'-'** a constant value; the mark **?** denotes an unpredictable effect depending of the particular test image and reference. The fact is that the ways P_1 affects M_2 and P_2 affects M_1 cannot be predicted. So, varying only these 2 parameters will already lead to totally unpredictable measurement changes. Far more complex is their dependance on all 6 parameters.

Here comes the configuration assistant ! It is used as follows.

At beginning of a new production run, a selection of good and bad markings is chosen as a learning set. Each component of **P** is given a range of possible values resulting in a subspace $\mathbf{P}^*$ of **P.** The automatic configuration, over $\mathbf{P}^*$, of the marking inspection system is then launched. It operates with one of the search methods described in 3.2 and with the objective functions defined previously (14), (15), (16). The automatic configuration search stops after a given amount of time and delivers the best configuration to be used for the marking inspection. The system is then ready to inspect the produced devices.

5. EXPERIMENTS

Experiments were conducted with several data sets and varying number of configuration parameters. All the search methods were tested long enough to gather statistically significant experimental results.

5.1. Data

The images of the devices were taken by standard RS170 monochrome imagers in our labs and in the production line of a customer. In the production line experiment, samples of "clean" markings were selected to make the Ω_x subsets whereas Ω_o subsets gathered really rejected printed markings. In the experiment, due to the unbalanced number of good versus bad provided markings at disposal, we had to "create" some bad devices. Synthetic defects are defects that were artificially done on good markings. These voluntary defects may take the form of scratches, written over, false marking, white/black tape stripes pasted over the marking.

3 data sets are considered as described in table 2:

	"74X" data set	"SOT" data set	"Casablanca" data set
Origin	Lab	Lab	Production line
Type of defect	Synthetic	synthetic	real
# of parameters	3+2	4+2	5+2
# of samples in Ω_x /Ω_o	10 / 10	21 / 36	20 / 20

Table 2: Data sets used

In this table, the number of parameters refers to the parameters automatically adjusted by the configuration assistant. These parameters are given possible discrete values belonging to a range; typically 10-20 allowed values per parameter. The 2 additional parameters are the decision thresholds T_i.

Figure 11 displays images of one “synthetic” defective, one good and one “real” defective devices.

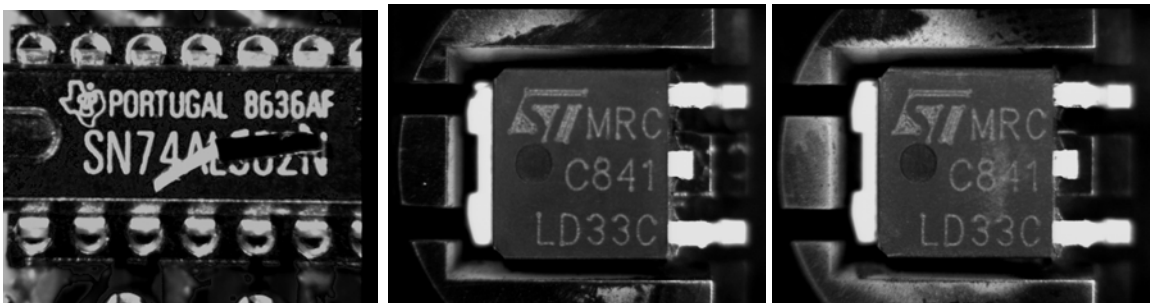

Figure 11: Images from inspected devices

5.2. Nature of Objective Function

The objective function measured with the Casablanca data set is plotted in figure 12 with 2 varying parameters. It is typical of the functions that were encountered in this work: it exhibits the three following characteristics. First the function takes zero value over the greatest part of the space. Secondly, the places where it takes non-zero values mostly form some connected regions. Finally the objective function is rather “regular” over these interesting regions. Thus, it can be expected that intelligent search heuristics will take advantage of these regularities of the objective function.

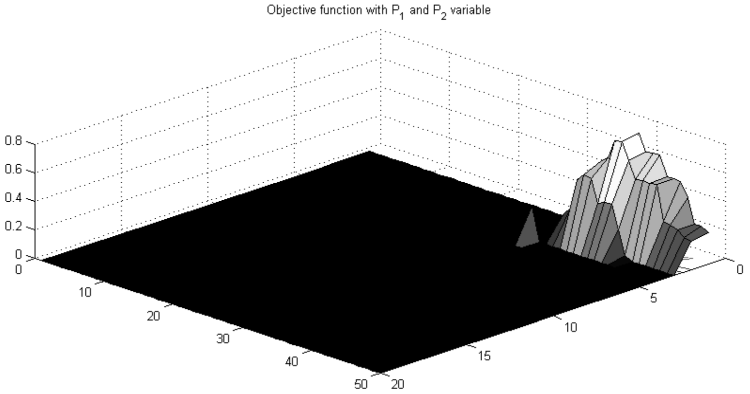

Figure 12: Objective function F for Casablanca data

5.3. Search Methods Comparison

A score is defined for further performance comparison of search methods. Given the objective function F(P), we define the normalized objective function as :

$$F_n = \frac{F}{\max\left[F\left\{\vec{P}\right\}\right]}. \tag{19}$$

and call **objective score** of a search the value of the highest F_n encountered during the search.

To practically compute the objective score, an exhaustive search over the possible configurations was first performed to gather the complete information about F(P). Of particular interest is the global maximum of F over {P} used in (19).

In order to compare the performance of the search methods, the mean objective score is plotted versus time (number of trials). A mean has to be used because of the intrinsic randomness of the used search methods, so only statistical results are significant. Figure 13 represents the mean objective score for the 3 search methods. The mean is computed over 100 trials. Results of a random search method are also displayed for comparison. The figure shows that the intelligent search methods GA and SA significantly outperform random selection. *Enhanced random* is less efficient.

This plot gives also an indication of the acceleration that a method brings for the configuration of the system. With the intelligent search methods, the configuration becomes nearly optimal (normalized score>0.95) already after only a small number of iterations. Typical speedup of SA or GA methods against random method are between 5-20 when approaching near optimal solutions. The speedup factor when compared with an exhaustive search of all possible configurations is important (about 100 for 5 parameters).

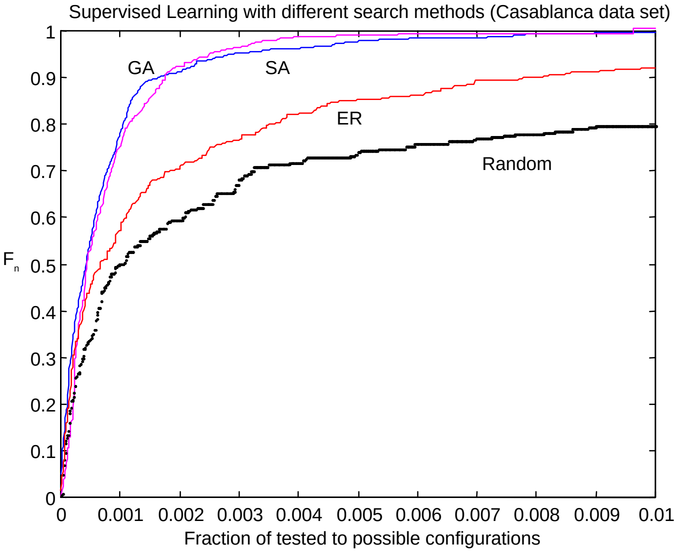

Figure 13: Mean objective scores of the search methods

The outstanding robustness of configurations obtained with the SA algorithm must be pointed out. This comes from the nature of the encountered objective function and from its gradient descent scheme that increase the probability to propose settings whose neighborhoods are also good settings.

5.4. Evaluation of the configuration

Finally a check of the real quality of the found configuration is done by observing the measurements for the learning set.
The working configuration assistant will not of course make the exhaustive search, it will be given a time period to perform the setting and it must find the best configuration during this period. The search method used here, is *simulated annealing*. To test the reliability 100 trials were performed each with a time period of 5 minutes, corresponding to approximately 3000 marking inspections. These inspections represent only 0.15% of inspections required to perform an exhaustive search. According to the objective score, the median and worst settings are used on the learning set's devices. Left half of figure 14 shows measurements with median-setting and right half with worst-setting. Good and bad subsets are separated by a clear margin even with the worst setting.

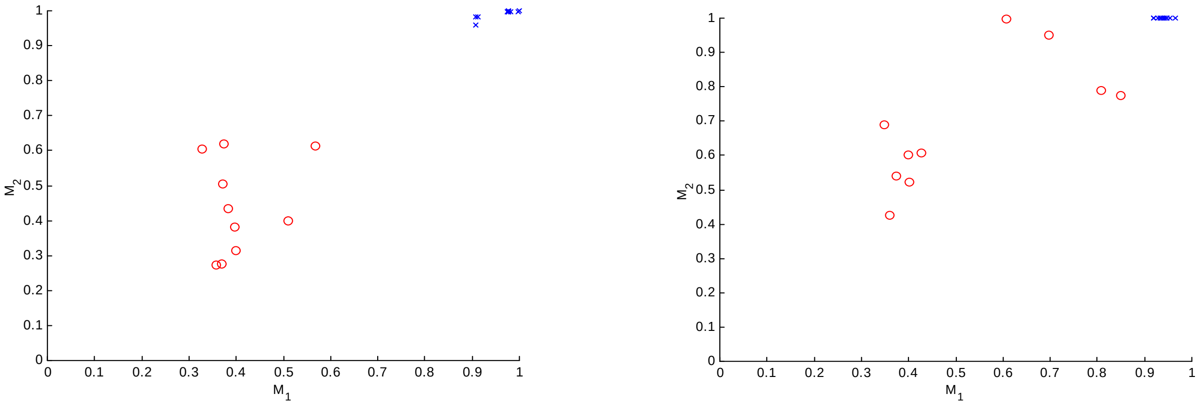

Figure 14: Learning set measurements with two different parameter configurations

6. CONCLUSION

This paper presented a configuration assistant for industrial visual inspection systems. With the help of carefully chosen objective functions and robust search methods, automatization of *and-classifier* based systems was achieved. This automatization relieves the human operator of the tedious setup of the inspection system and accelerates the process. The retained search methods, *genetic algorithms* and *simulated annealing*, showed to be typically much faster than a random search and drastically less time consuming than an exhaustive search.

Another advantage of the configuration assistant is the improvement in reliability. As the setup is performed with an objective function based on measurements instead of the classification results, a confidence value of the setup exists in the form of the objective score. This value leads to a more trustable setup and gives new information to the operator.

Future work will pursue and develop the presented method to other classifiers, the extension to multiclass decision systems being of interest. The various nature (continuous, discrete, ordered) of possible configuration parameters shall also be examined with respect to the search heuristics. The extension from a static to a dynamic configurator that learns during operation is a further subject of interest.

ACKNOWLEDGEMENTS

This research work was supported by the Swiss Commission for Technology and Innovation under project nr. 3460.1 . It was realized in collaboration with *Analytec SA*. Particular thanks go to Dr. Bernd Sommer, Dr Bernhard Schwab and Dr. Max Meyer.

AUTHORS

Olivier Hüsser received the degree of Physic-Electronics from the University of Neuchâtel, Switzerland, in 1995. He did a diploma work on character recognition and artificial neural networks. Since 1995 he worked as a research assistant in the pattern recognition group of the Institute of Microtechnology, Neuchâtel. His main research interests are image analysis, optimization processes and artificial intelligence.

Heinz Hugli is an assistant professor and research director at the Institute of Microtechnology, University of Neuchâtel, Switzerland and head of the Pattern Recognition Lab. This lab is active in computer vision with emphasis on industrial inspection, 3D object modeling and recognition. Heinz Hügli received the MS in 1971 and the Ph.D. degree in 1980, both from the Swiss Federal Institute of Technology in Zurich. He spent five year with the Image processing research group, Swiss Federal Institue of Technology and two years with the Medical Imaging Science Group, University of Southern California. In 1982 he joined the University of Neuchâtel where he was envolved in speech recognition and then founded the Pattern Recognition Lab. He is teaching at the University of Neuchâtel.

REFERENCES

1. N.Ueda, K.Mase, “Tracking Moving Contours using Energy-minimizing Elastic Contour Models”, *International Journal of Pattern Recognition and Artificial Intelligence*, 9 (3), 465-484, (1995).
2. N.S.Friedland, A. Rosenfeld, “Compact Object Recognition using Energy-Function-based Optimization”, in *IEEE Transactions on Pattern Analysis & Machine Intelligence (PAMI)*, 14 (7), 770-777, (1992).
3. S.Geman, D.Geman, “Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images”, *Proceedings of IEEE Transactions on Pattern Analysis & Machine Intelligence (PAMI)*, 6 (6), 721-741, (1984).
4. P.J. Besl and N.D. McKay, “A Method for Registration for 3-D Shapes”, in *Proceedings of IEEE Transactions on Pattern Analysis & Machine Intelligence (PAMI)*, 14 (2), 239-256, (1992).
5. B.T.Daniel, M.Cardew-Hall, “Geometric Variations : Analysis, Optimization and Control”, *International Journal of Pattern Recognition and Artificial Intelligence*, 11 (1), 79-113 (1997).
6. V.N.Vapnik, “A Training Algorithm for Optimal Margin Classifier”, in *Proceedings of the 5th annual ACM workshop on computational learning theory*, 144-152 (1992).

7. D.E.Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addislon Wesley, Reading Massachussets, (1989).
8. M.Srinivas, “Genetic Algorithms : a survey”, *IEEE Computer*, June 1994.
9. R.O.Duda, P.E.Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, Menlo Park, California, pp. 88-95, (1973).